

Комп'ютерний практикум № 3. Шаблони проектування. Поведінкові шаблони

Мета:

- ознайомитися з основними шаблонами проектування, навчитися застосовувати їх при проектуванні і розробці ПЗ.

Теоретичні основи

Шаблон проектування або патерн (англ. Design pattern) в розробці програмного забезпечення - повторювана архітектурна конструкція, що представляє собою рішення проблеми проектування в рамках деякого часто виникаючого контексту.

Патерн - це не закінчене архітектурне рішення, яке можна безпосередньо перетворити в вихідний або машинний код. Це опис підходу до вирішення проблеми, який можна застосовувати в різних ситуаціях.

Це опис взаємодії об'єктів, класів, адаптованих для вирішення загальної задачі проектування в конкретному контексті.

Патерни - це прості приклади, які показують правильні способи організації взаємодії між класами і об'єктами.

Патерни описують:

- Правильні способи формування внутрішнього стану (полів) і поведінки (методів) об'єкту або класу
- Правильні способи створення об'єкта (через виклик конструктора або іншим способом)
- Правильні способи об'єднання об'єктів всередині групи
- Правильні способи організації інформаційних потоків (виклик методів і черговість викликів), що дозволяють налагодити гармонійну взаємодію між об'єктами і групами цих об'єктів в об'єктно-орієнтованих системах

Зазвичай шаблон не є закінченим зразком, який може бути прямо перетворений в код. Це приклади розв'язання задач, які можна використовувати в різних ситуаціях. Об'єктно-орієнтовані шаблони показують відносини і взаємодію між класами або об'єктами, без визначення того, які саме кінцеві класи або об'єкти додатки будуть використовуватися.

Будь-який шаблон проектування може стати палицею з двома кінцями: якщо він буде застосований не до місця, це може обернутися катастрофою і створити багато проблем в подальшому. У той же час, реалізований в потрібному місці, в потрібний час, він може стати справжнім рятівником.

Існує три основних види шаблонів проектування:

- **Породжуючі патерни.** Це патерни, які абстрагують процес породження класів і об'єктів.
 - Абстрактна фабрика (Abstract Factory)
 - Будівник (Builder)
 - Фабричний метод (Factory Method)
 - Прототип (Prototype)
 - Одинак (Singleton)
- **Структурні патерни.** Розглядають, як класи і об'єкти утворюють більші структури - більш складні за характером класи і об'єкти.
 - Адаптер (Adapter)
 - Міст (Bridge)
 - Компоновщик (Composite)
 - Декоратор (Decorator)
 - Фасад (Facade)
 - Пристосуванець (Flyweight)
 - Заступник (Proxy)
- **Поведінкові патерни** - вони визначають алгоритми і взаємодію між класами і об'єктами, тобто їх поведінку.
 - Ланцюжок обов'язків (Chain of responsibility)
 - Команда (Command)
 - Інтерпретатор (Interpreter)
 - Ітератор (Iterator)
 - Посередник (Mediator)
 - Зберігач (Memento)
 - Спостерігач (Observer)
 - Стан (State)
 - Стратегія (Strategy)
 - Шаблонний метод (Template method)
 - Відвідувач (Visitor)

Постановка задачі комп'ютерного практикуму № 3

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

- 1) Вивчити поведінкові патерни. Знати загальну характеристику та призначення кожного з них, особливості реалізації кожного з поведінкових патернів та випадки їх застосування.
- 2) Реалізувати задачу згідно варіанту, запропонованого нижче. Розробити інтерфейси та класи з застосування одного або декількох патернів. Повністю реалізувати методи, пов'язані з реалізацією обраного патерну.
- 3) Повністю описати архітектуру проекту (призначення методів та класів), особливості реалізації обраного патерну. Для кожного патерну необхідно вказати основні класи та їх призначення,
- 4) Навести UML-діаграму класів

Варіанти індивідуальних завдань:

- 1) Реалізувати алгоритм гри sudoku. Реалізувати можливість «взяти назад хід».
- 2) Реалізувати задачу обрахунку кількості грошей, котрі завантажив відвідувач, в термінал по прийому грошей. Передбачити, що окремо можливість приймати монети та банкноти. Крім того, для монет різного номіналу існують окремі прийомники. Виводити загальну внесену суму та кількість монет та банкнот кожного номіналу окремо.
- 3) Реалізувати задачу моделювання роботи банкомату. Передбачити, що користувач може ввести невірний PIN-код, в банкоматі може не вистачити грошей для виконання операції користувача, банкомат може бути заблокований (в ньому можуть бути відсутні гроші), на картці користувача існує ліміт на суму, котру користувач може зняти за один раз.
- 4) Реалізувати ігровий простір, що подає персонажа з наступними необхідними атрибутами (позиція персонажу, склад артефактів, рівень "здоров'я" тощо). Реалізувати механізм збереження/встановлення стану персонажа.
- 5) Реалізувати бібліотеку для пошуку зразків рядків (регулярних виразів). Повинні підтримуватись операції: пошук рядків по точному співпадінню, пошук рядків по неточному співпадінню, застосування оператора «ТА (AND)», застосування оператора «АБО (OR)».
- 6) Розробити модель ігрової системи. Передбачити наявність фентезійних персонажів: орки, тролі, пегаси, ельфи, вампіри, гарпії і ін. Врахувати, що деякі персонажі ходять, інші - літають, треті - і ходять і літають. Літати також може група персонажів за допомогою магії.
- 7) Реалізувати задачу контролю руху літаків диспетчерами. Самі літаки напряму не взаємодіють один з одним. Повинно бути мінімум 4 взлітно-посадкових полос та 4 термінали. Одночасно на кожній полосі та у кожного терміналу може знаходитись тільки один літак.
- 8) Реалізувати арифметичні і логічні операції для комплексних чисел

- 9) Реалізувати модель, в якій існує лінійний список об'єктів та реалізується можливість послідовного обходу у прямому та зворотному напрямках оминаючи порожні елементи цієї структури та не розкриваючи її сутності перед користувачем.
- 10) Розробити систему Поштове відділення. З видавництва в поштове відділення надходять газети і журнали. Поштове відділення відправляє отримані друковані видання відповідним передплатникам.
- 11) Розробити модель вибору способів сортування та пошуку максимального / мінімального значення масиву числових об'єктів.
- 12) Реалізувати прототип графічного редактору простих геометричних фігур з функцією покрокового поновлення стану та повернення змін.
- 13) Розробити модель системи відслідковування курсів валют та біржових котирувань в режимі реального часу.
- 14) Замовлення на отримання гранту для навчання може перебувати в кількох станах: створений, розглядається, відкладений, відхилений, підтверджений, відкликаний та інше. Визначити логіку зміни станів і розробити модель системи.
- 15) Реалізувати модель системи ведення замовлень в інтернет-магазині. Повинні бути реалізовані різні стани замовлення (прийнятий, в обробці, оплачений, відмінений, дозволено відвантаження, дозволена доставка, доставлений в точку видачі, покупець забрав замовлення). Про кожну зміну статусу необхідно інформувати покупця. Передбачити, що в моделі задіяні ще комірник та кур'єр.
- 16) Розробити модель системи, що реалізує елементи структури комп'ютера (процесор, пам'ять, відеокарта тощо). Реалізувати механізм додаткових операцій над структурою комп'ютера без зміни її елементів. В якості ілюстрації такого механізму розробити операцію визначення потужності, що потребує комп'ютер.
- 17) Реалізувати задачу перевірки реєстраційних даних покупця інтернет-магазину (ПІБ, телефон, e-mail, адреса доставки, тип доставки, відділення пошти (при необхідності)). Необхідно забезпечити наступні види перевірки: коректність ведених даних по кожному елементу, наявність SQL-ін'єкцій, використання XSS-вразливостей.
- 18) Реалізувати модель системи розбору алгебраїчних виразів з операціями +, *, /.
- 19) Завдання, яке виконується студентом, може перебувати в станах: видано, виконано, здано на перевірку, перевірено, прездати на перевірку, не виконано. Визначити логіку зміни станів і розробити модель системи.
- 20) Проходження платежу через банківську систему супроводжується цілою низкою дій: фіксуючих, контролюючих, такі що знімають відсоток банку та інші вирахування і дії. Побудувати ланцюжки для різного виду

платежів (звичайних, пільгових, державних, внутрішньобанківських) відповідно до предметної області та розробити модель системи.

- 21) Реалізувати алгоритм гри «хрестики-нулики». Реалізувати можливість «взяти назад хід».
- 22) Реалізувати завдання прокату обладнання. При цьому необхідно розраховувати вартість обслуговування в залежності від способів і умов прокату. Способи та умови можуть бути стандартними, пільговими і штрафними. Можлива поява нових умов. Необхідно реалізувати конкретні алгоритми розрахунку для кожного поєднання пропонованої послуги і способу її придбання.
- 23) Реалізувати завдання конвертації зображення, завантаженого користувачем з заданої адреси, в обраному користувачем форматі (.jpeg, .gif, .png)
- 24) Реалізувати задачу виконання ходу шаховою фігурою.
- 25) Визначити специфікації класів, що реалізують елементи мережевої структури. Реалізувати модель структури (кабель, сервер, робоча станція). Реалізувати механізм додаткових операцій над мережевою структурою без зміни її елементів. В якості ілюстрації такого механізму розробити операцію визначення кошторису такої структури.